

應用網路服務實現開放式服務 與系統整合之探討

Using Web Services for Open Service and Integration

余顯強*

Shien-Chiang Yu

摘要

網路服務主要是透過標準的協定與方法提供簡易和標準化的分散式交易處理，執行在網路上的存取服務。其不僅可應用在圖書館現有資訊系統之間的整合，亦可提供開放式數位圖書館的服務需求。本文嘗試參考網路服務的服務模式，探討其整體應用的架構與相關標準，以及實際應用上所面臨的問題。

Abstract

A Web Service is a distributed application component based on the interchanging XML-encoded messages. Through standard protocols and methods, it provides distributed transaction in simplification and standardization. This Web Service can be used in the integrated information system as well as the service requirements of Open Digital Library. The purpose of this article is to tackle the problems of web services architecture, its associated standards, and issues of application.

關鍵詞：數位圖書館、應用程式整合、網路服務、延展式標示語言、簡易物件存取協定

Keywords: Digital Library; Application Integration; Web Services; XML; SOAP

一、前言

傳統圖書館邁向數位圖書館，強調的是整合性服務與典藏數位化的工作。圖書館在經過多年的自動化發展過程，在圖書館內常常會有許多為了不同目的而建置的資訊系統，如圖書目錄管理系統、期刊論文全文或目次系統、線上資料庫等等。但是，由於圖書館內部的資訊系統建置的來源與時期不一，這些系統使用的作業系統平台、程式開發語言、訊息傳遞標準，甚至使用的內碼都可能不同，因此在系統整合上，便遭遇許多的困難與問題。面對這些異質系統之間整合的問題，許多圖書館採取的主要方式是汰換多個舊有系統，再另外重新開發或採購新

* 世新大學資訊傳播學系助理教授(Assistant Professor, Department of Information and Communications, Shih Hsin University)

的系統，或是擴增系統的模組來取代其他系統，以達到整合的目的。

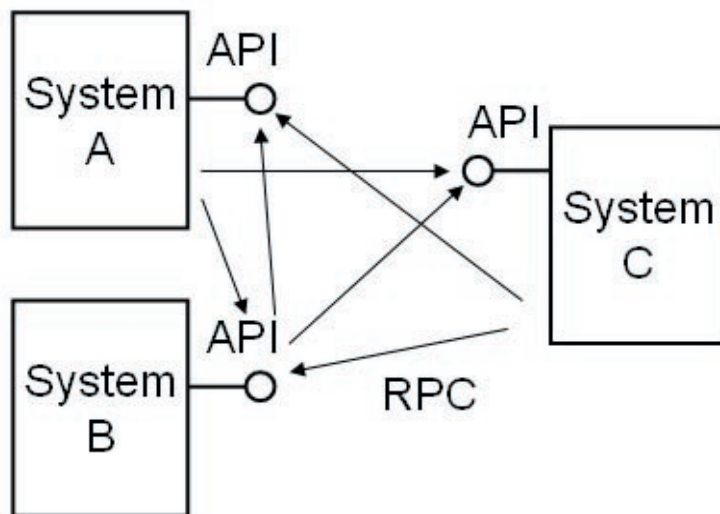
圖書館內部系統整合的方法通常主要包括網路整合、資料整合、移轉開放式系統平台、資料庫整合、系統架構整合等。但由於經費限制、使用習慣、人力因素等問題，而且新的系統通常也需經歷一段磨合期，因此，許多單位的系統並無法透過新舊系統替換方式來整合，而採用以資料傳遞模式或應用程式介面(Application Programming Interface, API)呼叫等仲介方式，間接達成資訊系統整合的方式。但是傳統上，透過資料傳遞的方式，並無法達到即時性的資料處理需求，而透過系統之間API呼叫的方式，又增加系統維護的複雜度、穩定性。另外，由於電腦網路安全日益受到駭客入侵、阻絕服務(Denial-of-service)、特洛伊(Trojan)木馬病毒的威脅，許多圖書館透過防火牆的架設，以保障館內自動化系統與資料的安全，

如此卻使得應用程式因為防火牆的阻擋，無法提供其他單位的系統透過遠端程序呼叫(Remote Procedure Call, RPC)執行圖書館內部系統的API。

針對上述資訊系統整合的問題及應用XML的彈性，而發展出網路服務(Web Services)的系統服務模式。Web Services主要是執行在web上的存取服務，透過標準的協定與方法提供簡易和標準化的分散式交易處理。Web Services不僅可應用在現有資訊系統之間的整合，亦可提供未來網路上公開與開放式的服務需求。本文嘗試參考網路服務的服務模式，探討應用在整合圖書館資訊系統的可行性與限制。

二、源起

基於XML支援語言中立(language neutral)的定義和平台中立(platform neutral)的特性，自動化系統都能夠輕易地修改符



圖一：傳統整合方式，透過開放API方式提供遠端系統呼叫

合XML交換資訊的格式(註1)，使得XML成為現今在資訊系統整合上主要採用的技術，包括提供訊息傳遞、資料格式、自動化營運流程等資訊流的主要媒介。企業資訊整合(Enterprise Application Integration, EAI)，便是應用XML作為訊息傳遞的基礎，其主要的作業模式是在各種不同程式語言、作業系統平台，或是後端異質的資料庫應用系統，採用XML作為彼此之間訊息往來的標準交換格式，再藉由XML訊息封裝的遠端程式呼叫來要求其他應用系統提供所需的服務或回應所需的資料，達成單一執行入口(portal)的資訊系統整合目的。

邁向數位圖書館的過程，圖書館自動化系統在應用層面的發展，可以分為下列三個階段：

- (1) 第一階段為自動化系統萌芽期，此時期的自動化系統純粹以書目資料處理為建置目的。
- (2) 第二階段為自動化系統擴充期，此時期的自動化系統逐步朝向資料庫管理和分析，並擴展至期刊目次全文等線上資訊需求服務。
- (3) 第三階段為自動化系統成熟期，此時期的圖書館自動化朝向主動式個人化的服務模式，服務的產生主要是依據使用者個人的資訊需求而定。數位典藏的形式不僅只有MARC書目型態的資料，強調的是結合各種metadata的管理與應用。

另外，結合多媒體資料的呈現方式也使圖書館處理的資料型態更加多樣化了。為了

提供使用者透通的(transparent)檢索需求，不僅圖書館內部必須整合既有各種自動化系統，館際之間的服務，如館際互借、開放式檢索等服務亦需整合。整合的優點可以縮短與館際之間或合作廠商間的交易流程與時間、節省內部人員的作業時間、介面管理一致性。但是，如何將各種不同系統平台 and 不同實作(implement)技術的功能，整合在單一網路瀏覽器的使用者介面之下操作，卻是一個非常困難的問題。

因此藉由XML作為標準資料的封裝技術，讓資料的交換跨越了各種平台、作業系統、程式語言和開發工具。使用XML封裝交換資訊的標準所發展出來的簡易物件存取協定(Simple Object Access Protocol, SOAP)技術，並且使用這種技術來呼叫實作技術的服務標準，如此一來不論遠端其他系統的實作技術是使用何種程式語言開發的物件，或是只是單純的CGI應用程式，用戶端都可以使用SOAP功能來呼叫這些實作程式碼，因此可以很簡單的整合網際網路上各種不同的實作技術。上述即為網路服務(Web Services)的基本概念，亦是Web Services所要解決的問題。

三、網路服務整合

1. 何謂網路服務

傳統的軟體工程設計重點在結構化與模組化設計，透過模組化的概念可以將資訊系統分組各個擊破(Divide and Conquer)。從模組化設計發展到物件導向開發，引進

了繼承(inheritance)、封裝(encapsulation)與多形(polymorphism)等技術，並使得系統發展的方法論也朝此方向發展。系統物件化的概念，配合應用XML為資料描述的SOAP訊息封裝標準，因此產生了網路服務(Web Services)的應用機制。主要從事網路服務基礎建設的提供廠商IBM，針對Web Services的定義如下(註2)：

Web Service是指用來描述透過XML訊息，並在網路環境上存取的一組操作(operations)的介面。一個Web Service使用一個稱之為服務描述的XML標示，用來描述所有與伺服器互動的細節，包括訊息格式、傳輸協定和位址。介面隱藏了實作的服務細節，允許其能夠運行在獨立的硬體或軟體平台，以及各種不同程式語言所開發的系統上。使提供Web Services的應用程式能夠達到鬆散偶合(loosely coupled)、元件導向(component-oriented)、橫跨技術等能力。Web Service能夠獨立或與其他系統所提供的Web Services配合，完成實現一項或一組的特殊複雜的聚合作業或企業交易。

而主導許多軟體開發規範的Microsoft對Web Services的定義為(註3)：

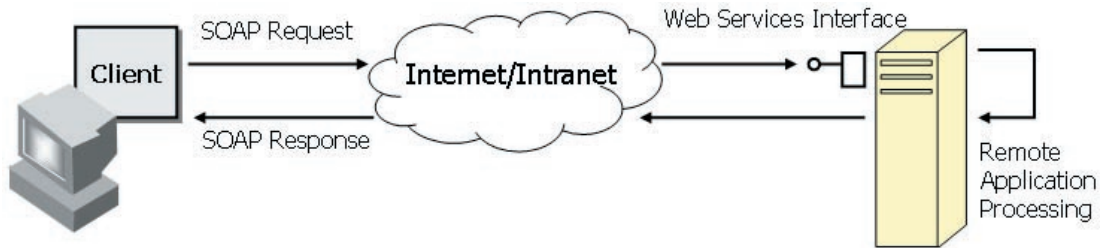
Web Services是用來提供給其他應用程式資料和服務的一組應用邏輯。應用程式透過無所不在的網路協定，如HTTP、XML和SOAP存取Web Services，而不需在意每一個Web Service是如何實作的。

另外，Microsoft還有針對Web Services技術上比較詳細的解釋(註4)：

一個Web Service是使用標準網際網路存取的可程式化應用邏輯。Web Services結合了最佳的元件發展和網路應用的觀點。Web Services就像是黑箱作業一般，提供在應用上無須顧慮服務的實作技術。不像現在的元件技術，Web Services不需透過物件模式規格(object-model-specific)，如Distributed Component Object Model(DCOM)、Remote Method Invocation(RMI)、Internet Inter-ORB Protocol(IIOP)等協定存取，而是透過無所不在的網路協定和資料格式標準，例如：HTTP和XML。

簡而言之，Web services只是定義了應用系統如何透過XML封裝並傳送資訊、註冊並描述這些服務、系統之間如何提出要求 and 回應的一組標準協定(註5)。不過，Web Services並非一定是發展在全球資訊網路(WWW)上的應用服務。一個Web Service能夠應用在任何形式的區域網路、網際網路(Internet)，或是企業網路(Intranet)內(註6)。此外在建構應用程式時，也可將整個系統分為獨立的區塊，彼此間透過Web Services互動。例如建立一個分散式數位圖書館，應用Web Services提供分散式查詢、驗證、館際互借、檔案傳遞、圖書採購付款等功能。這些Web Services能夠依據不同使用社群的需求，而結合成一個實際的數位圖書館應用環境。事實上，Web services架構可以被想像成一個提供服務而非提供資訊的數位圖書館(註7)。

2. 基本所需技術



圖二：Web Services應用SOAP在網際網路上運作的流程

事實上，Web Services的實作非常簡單，Web Services使用的基本技術大多是目前已經被廣泛接受的開放標準。包括JAVA與Microsoft的.Net及各主要軟體開發工具均支援Web Services的軟體元件，提供簡易的方式實作Web Services。因此圖書館欲將現有系統增加Web Services的服務功能時，只是使用現有的技術標準，再加上其原本系統所使用的特定語言或是平台的功能，開發符合SOAP訊息交換的物件即可，而這些程式可以依靠許多軟體廠商的開發工具便可輕易地達成。

以整體應用觀點來看，Web Services主要是應用在系統整合上的方法，其主要是將一個組織內資訊系統的功能整合，或各單位、館際之間資訊系統的整合。資訊系統藉由開放出Web services，提供其他無論是在不同硬體平台上或不同程式語言所開發的資訊系統來存取，以達到功能分享與整合的目的。將系統複雜的處理行為與回應結果，透過SOAP訊息協定溝通，使得傳統圖書館系統一對一的整合模式，可以藉由Web Services達成一對多的系統整合。

圖二說明SOAP和Web Services在網際網

路中運作的原理。使用者前端藉由SOAP封裝需求的資訊，透過網際網路傳遞到後端的伺服器(server)，再藉由Web Services的服務介面呼叫後端的功能，最後將取得的資訊以XML封裝成SOAP，傳遞回使用者前端。由於SOAP是實作在HTTP通訊協定上，也可呼叫到任何支援HTTP的平台，並且能夠輕易的穿越防火牆，因此克服現今許多應用軟體元件無法完成的工作。

Web Services藉由Internet/Intranet實際動態地發佈、搜尋並執行服務，其利用的技術標準還在陸續的制定中，目前較為成熟的標準如XML、服務描述語言WSDL(Web Services Description Language)、SOAP、註冊及儲存機制UDDI(Universal Description, Discovery, and Integration) (註8)等。透過這幾個技術標準讓服務提供者、服務仲介者與服務使用者彼此形成了需求、仲介與供給的服務關係，而這些技術標準在Web Services運作環境所扮演的角色分述如下：

(1) XML

Web Services若要讓異質平台之間不同的系統可以相互溝通和共用資料，則傳輸的資料必須要以一定的標準格式進行，而此一

標準格式便是XML。

(2) SOAP

不同系統之間發送和接收這些標準的XML資料，必須有一標準的協定以達到相互通信的目的。這個協定就是SOAP，直到2000年5月8日為止，SOAP規格已經由W3C發表到1.1版，而1.2版也正在草擬之中(註9)。SOAP是一個輕量化的協定，用來處理分權與分散式環境的資訊交換(註10)。SOAP是以XML為基礎的檔案傳輸格式，其作用為遠端呼叫與要求Web Services。也就是說，SOAP是運用XML來實行遠端程式呼叫的通訊協定，它把物件之間的請求和回應都封裝成XML來傳遞。現在主要用來傳輸SOAP物件的協定為HTTP，但SOAP規格並未指定一定要使用HTTP，包括SMTP、FTP或其他傳輸協定均可作為傳輸的依據，只是HTTP在現階段的效益上比較適合。

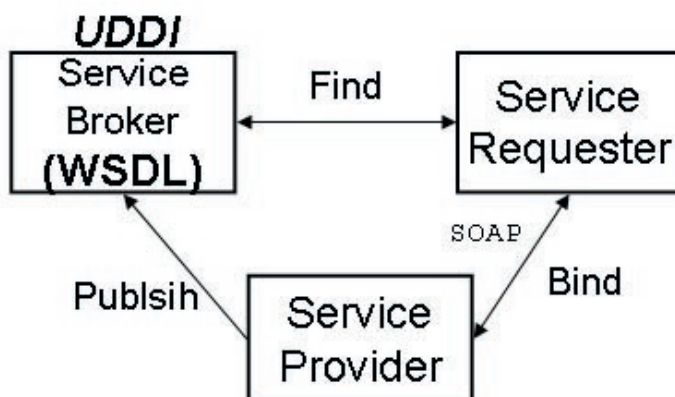
(3) WSDL

WSDL制定的目的是希望提供系統之間能夠知道彼此所提供的Web Services包含哪

些功能與服務。WSDL是一個以XML組成的檔案，這個檔案內容敘述了實作程式碼對外提供的函式說明，也就是各種可呼叫的函式名稱以及參數的資訊。當服務提供者欲對外公佈其所提供的Web Services，就必須以WSDL來建置一個描述的檔案。只要系統能夠取得特定一個Web Service的WSDL，就可以從其中瞭解它所提供的服務，以及如何呼叫這一個Web Service。

(4) UDDI

UDDI依據Web Services的特性分類成目錄，提供搜尋與整合的服務。將圖書館系統的Web Services註冊到UDDI上成為服務提供者，供其他系統呼叫使用，或是透過UDDI獲知Internet上有哪些適用的Web Services可供館內系統呼叫使用。UDDI類似於搜索網站的引擎或是電子商務的產品目錄搜索一般的作用，扮演服務仲介者的角色。Web Service的提供者必須在UDDI註冊中心註冊，而服務請求者若需要搜尋某項服務，便可到UDDI查詢。三者的關係如圖三所示(註11)：



圖三：服務提供者、服務請求者、服務仲介者的關係

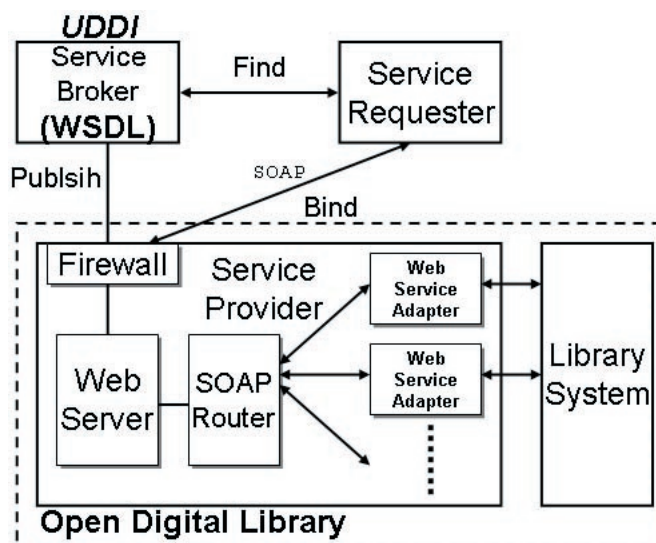
3. 整合特色與優點

Web Services對圖書館自動化系統間的整合提供了以下特色：

- (1) 鬆散偶合(loosely-coupled)的分散式結構。
- (2) 平台中立，與實作的開發工具、程式語言無關，低成本並可與圖書館內部現有應用程式整合。
- (3) 提供同步與非同步的程序呼叫。
- (4) 容易穿越防火牆。
- (5) 訊息處理模式以XML為基礎，與圖書館metadata的目標一致。

參考圖三的架構，因應數位圖書館開放性的應用趨勢，我們可規劃如圖四的開放性服務模式。圖書館內部建立了Web Services環境，除了可以透過Web Services整合內部既有的各個資訊系統，系統還可透過UDDI在Internet上找到有價值的Web Services，並展開Web Services之間的資訊連結與互動回

應，達成圖書館與圖書館，或圖書館與其他機構彼此之間系統的整合。舉例來說，圖書館負有知識的收集、典藏和讀者服務的任務，但在尚未達成開放式數位圖書館(註12)的今日，許多的資訊絕對無法完全透過地取得。但對於使用者而言，雖可以透過許多開放式的分散式檢索，如Z39.50、Open Archives Initiative(OAI)等協定，協助使用者向各資料提供者查詢或取得所需的文獻。但是上述這些協定主要是基於資料提供者與服務提供者關係之建立為基礎，彼此間必須設定相關的服務機制或表列，且服務的範圍主要是文獻、目錄或metadata的傳遞。而透過Web Services則可以提供圖書館藉由UDDI主動開放或找出網際網路上有利於使用者的服務，提供系統與系統之間館際互借、目錄分享、分散式檔案檢索等服務，並自動的展開系統間的互動，使「整合」有了新的詮釋與未來。



圖四：圖書館內部建立了Web Services環境，提供對內與對外的服務整合

就整體而言，採用Web Services具有下列六項優點(註13)：

(1) 單純化(Simple)

傳統系統整合方式所使用的技術大多是DCOM、CORBA，或是透過EDI的資料交換模式來達到整合。但就採用Web Services而言，一旦用來發展及使用Web Services的架構已建置完成，即可輕易建立一個橫跨多個應用系統的新運作流程，故Web Services在設計、發展、維護及使用上皆較為簡單化。

(2) 開放式的標準(Open Standard)

相較於傳統式的系統整合，Web Services所採用的標準為開放式標準，包含UDDI、SOAP、WSDL，及HTTP。均是目前已存在的通訊協定，使用單位無須再投資更多的成本來支援網路協定，此優點為Web Services可被廣泛使用的原因之一。

(3) 彈性(Flexible)

傳統的系統整合大多採用點對點(Point-to-Point)的整合方法，系統與系統會緊密結合在一起，一旦有所異動便會影響另一端，造成維護上的不便性；相較於此，Web Services所採用的整合方式，會透過使用被公佈出來的服務來與其他系統進行結合，充分保留系統與系統之間整合的彈性。

(4) 整合範圍廣(Scope)

就傳統式的系統整合而言，一個應用系統僅提供一個整合入口介面；但就Web Services而言，更可將應用系統區分成多個獨立的邏輯單位，亦提供多個整合的連結點。以訂單管理系統為例，Web Services的整合方式可提供接單、訂單狀態查詢、訂單

確認、收款及付款等服務作業系統整合的入口。

(5) 有效性高(Efficient)

鑑於Web Services的整合方式可以為單一應用系統提供多種整合入口的優勢，Web Services的整合方法較傳統方式更具有效性。

(6) 具動態性(Dynamic)

自動化系統整合的過程之中，若採用Web Services的方式來進行，則只需從服務提供者(Services Provider)所公佈出來的服務中，選擇適當的服務，並納入營運流程中，進而達到系統整合的目的。相較於傳統式的整合方法，Web Services是以動態整合介面來整合系統，較具動態性。

四、圖書館的應用

1. Portuguese National Library

The Portuguese National Library與書目供應商BookMARC公司，共同發展一個大型的「Joaquim Carvalho of BookMARC」計劃，應用Web Services提供不同系統之間，分享其上百萬筆的UniMARC書目紀錄。所提供的服務包括透過SOAP傳送適當的訊息便能取得所需的單一UniMARC書目紀錄、透過查詢各種不同的欄位取得所需的資料集合，以及包括搜尋、輸入、維護和驗證資料等服務功能(註14)。提高圖書館與書商之間書目資訊、交易資料的互通與分享效益，簡化彼此訊息溝通的管道。

2. ALADIN數位圖書館系統

ALADIN (Access to Library And Database Information Network)是由Washington Research Library Consortium (WRLC)負責發展的系統平台。其發展目標是能夠透過館際互借、和線上目錄，提供數位典藏的存取、資料庫和圖書目錄的訂購服務(註15)。此系統應用Web Services作為中介者的服務模式，透過單一且一致的介面提供許多服務元件(components)讓其他系統取得所需的資訊或服務。這些應用Web Services的元件包括(註16)：

- 讀者鑑別(authentication)
- 語音資料的存取與串流(streaming)控制
- Web文獻傳遞
- 資源尋找服務
- 內文傳遞的入口(portal)

3. Z39.50-International: Next Generation (ZING)計劃

ZING計劃現在主要執行的創始機構包括：SRW (Search/Retrieve Web Service)、SRU(Search/Retrieve URI Service)、ZOOM(the Z39.50 Object-Orientation Model)、ez3950、ZeeRex(Z39.50 Explain, Explained and Re-Engineered in XML)(註17)。此計畫的目標是希望使Z39.50能更廣泛且更容易地實作，其中SRW/U是希望能將Z39.50改良並發展成為開放式檢索協定的主流，而ZOOM的目標是希望能簡化Z39.50的複雜度，ez3950則是應用SOAP作為Z39.50前後端的執行媒介，而SOAP所封裝的訊

息則是使用XML編碼技術(XML Encoding Rules, XER)。

整體而言，ZING主要的方法便是使用Web Services，包括XML, HTTP, SOAP, WSDL等標準的web技術，以便建立一個輕量化的Z39.50資訊存取協定(註18)。

五、問題與討論

正如許多新技術的發展，是要解決現有許多實際系統應用上的問題。不過由於Web Services是透過現有網路協定而形成的元件服務機制，因此Web Services亦會面臨網路上許多存在的問題，如使用權限、追蹤服務的使用情況、紀錄被使用的服務，甚至使用的收費方式，都是需要被考慮處理的議題，茲討論如下(註19)：

1. 信賴度(Reliability)

Web Services是透過相互運作而形成的元件服務機制，所以在此環境下主要是資料相互傳遞、運算、記錄等動作。雖然在呼叫遠端的服務後會收到一回傳值，但無法獲知服務之間的運算方式與程式流程是如何進行的。舉例來說，某網站宣稱有資訊選粹服務(Selective Dissemination of Information, SDI)，使用者只要輸入一次電子郵件信箱，即可以申請多家圖書館的書目與期刊目次服務。假設有人故意建置一資訊選粹的服務，提供各圖書館資訊系統透過Web Services取得相關服務，但其目的是在收集各圖書館資訊系統傳遞而來的使用者信箱位址、個人資訊，以便能將資料出售牟利，如此將讓圖書

館與使用者皆會蒙受損失，而整個運作過程卻沒有具體的信賴機制來防範。

所以對於建置好的服務，若無法建立一稽核與鑑別機制，則服務也就沒有任何信賴度可言，相對的也就會阻礙Web Services的發展。

2. 安全性(Security)

安全在網際網路上是存在已久的議題，Web Services仍然必須面臨此一問題。在資料傳輸的過程當中，難保不會有駭客趁機擷取資料。服務經由WSDL的描述後，所有參數與回傳值的個數與型態均可輕易得知，駭客對於資料的分析與擷取將更為方便，所以需要考慮資料安全的防護措施。Web Services主要的重點在於系統和服務之間的溝通，所以身分認證的問題並不能只針對使用者個人，還需要對服務元件作身分的認證，但是服務的認證卻比人的認證更加困難。

3. 異動交易(Transaction)

交易的問題是Web Services最大的障礙之一。在Web Services的架構之中，資料極為可能被分散儲存與管理在不同的環境之中。當一個交易發生時，便會產生分散式的交易行為(Distributed Transaction)。分散式的交易基本上會有一些問題需要克服，首先是交易時間的問題：當一個交易產生時，必需能夠在限定的時間內完成，但因為Web Services是分散在網際網路之中，所以交易完成的時間會受到網路頻寬或遠端系統執行負荷的影響，假若交易時間設的太短，將造

成交易失敗的機率大增；但交易時間設的太長，又會影響到服務品質。

第二個是交易回復(Rollback)的問題，當交易失敗時，必須對此次交易所有異動紀錄進行回復，回復到交易未發生之前的狀態。傳統集中式交易機制的情況，交易的資源包含資料庫、檔案等均是統一管理，當有交易發生的時候，大多會以鎖住的方式避免其他的交易同時間改變資料的狀態。然而在Web Services的架構下，鎖住的作業方式會延誤服務回應的時間，而影響服務的效率。不採鎖住的策略又容易造成資料同步修改的問題。

4. 協定的先天限制(Protocol constraint)

(1) HTTP

透過HTTP的訊息傳遞方式有兩個主要問題：

- a. 不保證訊息封包一定送達接收端。
- b. 不保證訊息封包以發送端的發送順序到達接收端。

當網路頻寬壅塞時，訊息封包將可能會被丟棄，因此網路頻寬很明顯的會是Web Services的瓶頸。其所導致訊息傳遞的時間延遲、頻寬及訊息同步問題，都將成為影響Web Services系統執行效率的因素。

(2) SOAP

Web Services實際上就是採用SOAP方式來運作，因此SOAP在執行效率上的影響因素，也將會直接影響到Web Services的執行效率。SOAP採用XML做為訊息封裝的標準，無論是從SOAP中讀取出訊息封套

(SOAP Envelop)，或是剖析XML都需耗費時間。尤其是大部分的XML剖析器(Parser)必須支援型態檢查(data-type checking)、語法驗證(well-formed和validated)等功能，因此在程式大小、處理時間及其所需記憶體空間上都會耗費許多系統的資源。

另外，使用XML方式來儲存資料會比一般文字方式所需要的空間還大，導致訊息在透過網際網路傳遞的效率降低，因此訊息的大小也會是設計Web Services時須考慮的問題。除了訊息封包大小的問題之外，雖然國際上已有許多組織公開宣稱將會支援SOAP，不過在兩個分散式物件之間所傳遞的訊息內容細節方面，如何描述和解譯訊息綱要來呈現資訊，以及如何使SOAP很方便的應用在遠端程式呼叫與回應上，仍有許多尚未明確規範之處。

(3) UDDI

UDDI主要的功能，是為了提供Web Services可以透過UDDI搜尋到所有的服務，並從所得到的資料進行整合。依據此一目的，UDDI並不需要太多效能的要求。但是Web Services的發展趨勢，勢必會漸漸走向目錄服務自動整合的境界，也就是以動態的方式找尋更適合的服務進行整合。如此一來，UDDI搜尋引擎的效能、功能、穩定性等等，也將會成為Web Services發展的一大問題。

六、結論

Web Services提供了內部與外部網路(internet/intranet)平台中立的方式和分散式

處理的技術，使用標準的XML協定和格式來實現系統之間的應用服務。此外，Web Services不僅提供了鬆散的方式來結合各項服務，而且還能夠透過其快速、簡易的開發方式，以及發佈服務元件、獲知外部服務元件並動態結合的優點，提供更多特殊的服務模式。因此，Web Services除了能夠輕易地達成「需求/回應」的功能，對於圖書館系統也還能提供下列新的服務模式：

- (1) 圖書館內部各系統和合作廠商系統之間功能層面的整合。
- (2) 館際間服務的股份，包括開放式檢索、書目、館際互借等。
- (3) 主動搜尋網際網路內適當的服務，提供讀者各種服務的轉介機制。
- (4) 能夠依據不同使用者社群的需求，而結合成一個實際的數位圖書館應用環境。

註釋：

- 註1 Shien-Chiang Yu, and Ruey-Shun Chen, "Developing an XML Framework for An Electronic Document Delivery System," *The Electronic Library* 19:2 (2001): 103.
- 註2 Heather Kreger (2001), "Web Services Conceptual Architecture (WSCA 1.0)", <<http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>>
- 註3 XML Web Services, available: <<http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000442>>

- 註4 Mary Kirtland (Jan. 2001), "A Platform for Web Services", available: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnWebsrv/html/Websvcs_platform.asp>
- 註5 Roy Tennant, "What to Know About Web Services," *Library Journal* 127:12(2002): 32.
- 註6 Steve Graham and so on, "Building Web Services with Java," *SAMS* 2001: 10.
- 註7 Tracy Gardner, "An Introduction to Web Services," *Ariadne UKOLN* 29(September 2001). <<http://www.ariadne.ac.uk/issue29/gardner/>>
- 註8 Scott Seely, *SOAP: Cross Platform Web Service Development Using XML* (New Jersey: Prentice Hall PTR, Inc., 2002), 163.
- 註9 同註6, 頁31-32。
- 註10 Don Box and so on, "Simple Object Access Protocol (SOAP) 1.1," (May 2000) <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>>
- 註11 同註7。
- 註12 Hussein Suleman, and Edward A. Fox, "A Framework for Building Open Digital Libraries," *D-Lib Magazine* 7:12 (December 2001) <<http://www.dlib.org/dlib/december01/suleman/12suleman.html>>
- 註13 劉威遠、黃雯汝, 「Web Services帶來整合革命」, *資訊與電腦* 261期(民國91年4月), 頁19-25。
- 註14 同註5。
- 註15 Kyle Banerjee, "How Does XML Help Libraries," *Computers in Libraries* 22:8(September 2002): 30-34.
- 註16 Don Gourley, "An Architecture for the Evolving Digital Library," *Washington Research Library Consortium*(2001) <<http://www.educause.edu/ir/library/html/edu0122/edu0122.html>>
- 註17 The Library of Congress ZING: Z39.50-International Next Generation, The Library of Congress, (2002) Available: <<http://www.loc.gov/z3950/agency/zing/zing.html>>
- 註18 Leigh Watson Healy, and William E. Moen, "Z39.50: A Primer on the Protocol," *NISO Press* (May 2002): 9.
- 註19 余顯強, 陳瑞順, 「網路服務: 資訊系統整合的應用」, *中國圖書館學會會報* 68期(民國91年六月), 頁125-134。